Please replace the following Claims with the clean Claims presented here as follows. Marked-up Claims showing the changes made appear after the REMARKS section.

IN THE CLAIMS:

Claim 1 (Once Amended):

1. For use in an instruction processor that executes instructions included in a predetermined instruction set at an execution rate determined by a system clock signal, a synchronous instruction pipeline, comprising:

a pipeline execution circuit to process a first predetermined number of instructions simultaneously, each of said first predetermined number of instructions being in a respectively different stage of execution within said pipeline execution circuit, instructions being capable of advancing to a next stage of execution within said pipeline execution circuit at a time determined by the system clock signal; and

a pipeline fetch circuit coupled directly to said pipeline execution circuit to retain a second predetermined number of instructions simultaneously, each of said second predetermined number of instructions being in a respectively different stage of processing within said pipeline fetch circuit, an instruction being capable of advancing to a next stage of execution within said pipeline fetch circuit at a time determined by the system clock signal and independently of the times at which instructions advance to a next stage of execution within said pipeline execution circuit.

Claim 7 (Once Amended):

7. The synchronous instruction pipeline of Claim 5, wherein said pipeline execution circuit includes a microcode-controlled sequencer to control execution of extended stages of execution of extended-mode ones of the instructions, wherein during said extended stages of execution, ones of the instructions being executed by said pipeline execution circuit are not advancing to a next stage of execution within said pipeline execution circuit, and wherein said first selection circuit includes a control circuit to allow an instruction to enter said pre-decode stage of processing while said extended-mode ones of the instructions are not advancing to a next stage of execution within said pipeline execution circuit.

Claim 8 (Once Amended):

8. For use in an instruction processor that executes instructions of a machine instruction set, a synchronous pipeline system comprising:

a plurality of execution logic sections, each of said execution logic sections being coupled to at least one respective other one of said execution logic sections, each of said execution logic sections to perform a predetermined stage of execution of any of the instructions, and whereby each of said execution logic sections is capable of receiving a new instruction to process at predetermined time increments; and

a2

a plurality of fetch logic sections wherein at least one of said plurality of fetch logic sections is coupled directly to at least one of said plurality of execution logic sections, each of said fetch logic sections being coupled to at least one respective other one of said fetch logic sections, each of said fetch logic sections to perform a predetermined pre-execution stage of instruction execution, each of said fetch logic sections being capable of receiving a new instruction to process at said predetermined time increments and in a manner that is independent of whether any of said plurality of execution logic sections receives a new instruction to process.

Claim 14 (Once Amended):

14. For use in an instruction processor having a synchronous instruction pipeline that executes instructions at a rate determined by a system clock, the instruction pipeline including a predetermined number of execution logic sections coupled to each other in sequence, each to perform a respectively different stage of execution on any instruction, and a predetermined number of fetch logic sections coupled to each other in sequence, each to perform a respectively different stage of pre-execution on any instruction, a method of processing instructions, comprising the steps of:

a3

(a) providing at least one of the fetch logic sections that is coupled directly to at least one of the execution logic section;

(b) processing a respective one of the instructions by each of the execution logic sections for a first predetermined time period;

(c) allowing ones of the execution logic sections to each pass said respective one of the instruction to another coupled one of the execution logic sections after said first predetermined time period elapses;

(d) allowing at least one of the execution logic sections to retain said respective instruction for longer than said first predetermined time period; and

(e) allowing ones of the fetch logic sections each to begin processing a respective instruction during a subsequent predetermined time period that is subsequent to said first predetermined time period if said each of the fetch logic sections was not processing a respective instruction during said first predetermined time period.